

《C/C++ 学习指南》

第10.1讲：结构体的定义和基本使用

作者：邵发 QQ群：417024631
官网：http://www.afanihao.cn/c_guide/
答疑：<http://www.afanihao.cn/kbase/>

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

版权所有，侵权必究

引例1

下面的表格记录了多个联系人的相关信息：
每个联系人的信息：ID，名称，手机号

ID	姓名	手机号
201501	John	18601011223
201502	Jennifer	13810022334
201503	AnXin	18600100100
201504	Unnamed	13111011011

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

引例1：如何表示结构化信息？

使用之前学到的知识，可以表示为：
(注意选择合适的类型)

```
int id[4] = { 201501, 201502, 201503, 201504 };
char name[4][16] =
{
    "John", "Jennifer", "AnXin", "Unnamed"
};
char phone[4][16] =
{
    "18601011223", "13810022334",
    "18600100100", "13111011011"
};
```

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

引例1：如何表示结构化信息？

缺点：

- (1) 信息比较松散，不直观
- (2) 容易重名
- (3) 数据存取不方便

比如，现在要按ID来查找一个联系人，将联系人的信息返回。这个函数很难写出来，因为字段太多了。

```
int find (int id, char* name, char* phone)
{
}
```

貌似可以使用输出参数name, phone来返回联系人的信息。。

但是，如果字段变多，有100个字段呢？难道要用100个输出参数？

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

5

版权所有，侵权必究

引例1：如何表示结构化信息？

最好有一种新的数据类型，直接表示联系人所有属性。

```
Contact persons[4]; // Contact: 一种新的类型
```

```
Contact find(int id); // 直接返回一个Contact对象  
或
```

```
int find(int id, Contact* p); // 使用一个输出参数
```

这样的语法才是简洁直观的！

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

6

struct: 结构体 structure

C/C++的几个主要的基本类型，char/short/int, float/double, 以及数组，指针。但它们还不够用。。。

将基本类型组合起来，新形成了新的数据类型，称为“自定义类型”。

例：

```
struct Contact  
{  
    int id;  
    char name[16];  
    char phone[16];  
};
```

这里定义一个新的类型Contact，它的地位和int, double之类相同。

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

7

struct: 结构体

基本形式为

```
struct TypeName  
{  
    // Members  
};
```

其中，struct是关键字，TypeName为新类型的名称，Members则是一系列成员变量。

- ① TypeName的命名规则：和变量名的规则一样，见3.2讲。字母、数字和下划线的组合。
- ② 成员变量可以普通的变量/数组定义

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

8

版权所有，侵权必究

struct: 结构体

例如，

```
struct Contact  
{  
    int id;  
    char name[16];  
    char phone[16];  
};
```

定义一个新类型Contact，成员变量有id, name, phone

注意：成员变量的定义和普通普通的定义方式是一样的。

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

struct: 结构体

新类型的使用和基本类型差不多。。。

(1) 定义一个变量

```
Contact c;
```

(2) 定义数组

```
Contact cs[4];
```

(3) 指针

```
Contact* pc = &c;
```

(4) 作为函数参数

```
void test(Contact c);
```

(5) 作为返回值类型

```
Contact make(int id);
```

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

(1) 变量定义和初始化

定义一个变量

```
Contact a; // 定义一个变量，不给初始值
```

定义时初始化：用大括号将各个成员的初始值列出, 按顺序列表。（对比二维数组的初始化）

```
Contact a =
{
    201501,          // int的初始化
    "Jennifer",     // char数组的初始化
    "13810022334"
};
```

初始化时：

- ①注意使用大括号，末尾要加分号结束
 - ②各字段的初始值要与该字段的类型匹配
 - ③各初始值以逗号分开
- 按顺序初始化

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

(1) 变量定义和初始化

和一维数组一样，可以只初始化部分成员变量。

```
Contact a = { 201501, "John" }; //只初始化前2个成员
```

也可以直接清零

```
Contact a = {0};  
// 则该变量的每一位都是0
```

(1) 变量定义和初始化

在内存视图里，一目了然：各个成员紧密排列，相当于被组合在了一起。

```
Contact a =  
{  
    201501,    // int的初始化  
    "Jennifer", // char数组的初始化  
    "13810022334"  
};
```

观察a在内存窗口中的数据。。。

(2) 定义数组

可以和基本类型一样，定义数组，表示并排的多个对象。

```
Contact cs[4];
```

或定义时初始化，最后一个元素/字段后面不需要加逗号

```
Contact cs[4] =  
{  
    {201501, "John",    "18601011223"},  
    {201502, "Jennifer", "13810022334"},  
    {201503, "AnXi",    "18600100100"},  
    {201504, "Unnamed", "18601011223"}  
};
```

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

(3) 对成员的访问

使用点号访问其成员：

```
Contact a;  
a.id = 201501;  
strcpy(a.name, "John");  
strcpy(a.phone, "18601011223");
```

注：strcpy函数用于字符串的拷贝，需要<string.h>

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

(3) 对成员的访问

数组方式

```
Contact cs[4];  
cs[0].id = 201501;  
strcpy(cs[0].name, "John");  
strcpy(cs[0].phone, "18601011223");
```

注：strcpy函数用于字符串的拷贝，需要<string.h>

(3) 对成员的访问

注意：

访问成员指的是访问“结构变量”的成员

```
Contact a;  
printf("id: %d \n", a.id);
```

千万不要写成

```
printf("id: %d \n", Contact.id); //错！！
```

Contact只是一个类型的名字而已。只要变量才对应内存。

小结

- (1) 初步学习用struct定义新类型
- (2) 新类型的变量定义方式和基本类型差不多相同（注意其初始化方式）
- (3) 成员的访问：使用点号加上成员的名称